

```

1
2
3 from a0_items import *
4 import schedule
5
6 import time
7
8 from datetime import date
9
10
11 #https://plainenglish.io/blog/5-ways-to-schedule-jobs-in-python-99de8a80f28e
12
13 def task_1():
14
15     print ("starting")
16
17     for x in range (5):
18
19         now = datetime.now()
20
21         print (now, flush=True)
22         time.sleep(3)
23
24
25
26
27
28
29 def run_task_LOOP (time_pause, run_count):
30
31     for x in range (run_count):
32
33         task ()
34
35         time.sleep(time_pause)
36
37
38 #run_task_LOOP (time_pause = 5, run_count = 5)
39
40
41
42 # https://schedule.readthedocs.io/en/stable/
43 def run_task EVERY (run_every, time_value, end_datetime, task):
44
45     match run_every:
46
47         case "s":
48             print ("run task every " + str(time_value) + " SECONDS")
49
50             task_schedule=schedule.every(time_value).seconds.until(end_datetime).do(task)
51
52             while True:
53                 schedule.run_pending()
54                 time.sleep(1)
55
56         case "m":
57             print ("run task every " + str(time_value) + " MINUTES")
58
59             task_schedule=schedule.every(time_value).minutes.until(end_datetime).do(task)
60
61             while True:
62                 schedule.run_pending()
63                 time.sleep(1)
64
65         case "h":
66             print ("run task every " + str(time_value) + " HOURS")
67

```

```

68         task_schedule=schedule.every(time_value).hours.until(end_datetime).do(task)
69
70     while True:
71         schedule.run_pending()
72         time.sleep(1)
73
74     case "d":
75         print ("run task every " + str(time_value) + " DAYS")
76
77         task_schedule=schedule.every(time_value).days.until(end_datetime).do(task)
78
79         while True:
80             schedule.run_pending()
81             time.sleep(1)
82
83     case "w":
84         print ("run task every " + str(time_value) + " WEEKS")
85
86         task_schedule=schedule.every(time_value).weeks.until(end_datetime).do(task)
87
88         while True:
89             schedule.run_pending()
90             time.sleep(1)
91
92
93 run_task EVERY (run_every= 's', time_value= 3, end_datetime="2024-10-10 15:08", task =
task_1)
94
95
96
97
98 def run_task EVERY_WEEKDAY_AT (run_every, time_value, task):
99
100
101     match run_every:
102
103         case "mon":
104             print ("run task every MONDAY")
105
106             schedule.every().monday.at(time_value).do(task)
107
108             while True:
109                 schedule.run_pending()
110                 time.sleep(1)
111
112         case "tues":
113             print ("run task every TUESDAY")
114
115             schedule.every().tuesday.at(time_value).do(task)
116
117             while True:
118                 schedule.run_pending()
119                 time.sleep(1)
120
121         case "wedn":
122             print ("run task every WEDNESDAY")
123
124             schedule.every().wednesday.at(time_value).do(task)
125
126             while True:
127                 schedule.run_pending()
128                 time.sleep(1)
129
130         case "thurs":
131             print ("run task every THURSDAY")
132
133             schedule.every().thursday.at(time_value).do(task)

```

```
134
135     while True:
136         schedule.run_pending()
137         time.sleep(1)
138
139 case "fri":
140     print ("run task every FRIDAY")
141
142     schedule.every().friday.at(time_value).do(task)
143
144     while True:
145         schedule.run_pending()
146         time.sleep(1)
147
148
149 case "sat":
150     print ("run task every sATURDAY")
151
152     schedule.every().saturday.at(time_value).do(task)
153
154     while True:
155         schedule.run_pending()
156         time.sleep(1)
157
158 case "sun":
159     print ("run task every SUNDAY")
160
161     schedule.every().sunday.at(time_value).do(task)
162
163     while True:
164         schedule.run_pending()
165         time.sleep(1)
166
167
168 #run_task EVERY_WEEKDAY_AT (run_every="mon", time_value="13:30", task=task_1)
```